

Tutorial

omicR

Windows and Linux

Berenice Talamantes Becerra
2-25-2021

Contents

| | |
|--|-----------|
| <i>Introduction</i> | 3 |
| <i>Installing omicR with graphical user interface</i> | 4 |
| Windows..... | 4 |
| <i>Running “omicR” with graphical user interface</i> | 6 |
| 1. Create FASTA files and input files for BLAST / filtering..... | 7 |
| 2. Download genomes from the NCBI..... | 8 |
| 3. Create NCBI database for BLAST+..... | 9 |
| 4. BLAST and filtering..... | 9 |
| 5. Additional filtering..... | 12 |
| <i>omicR for R Studio</i> | 13 |
| Running “omicR” in R studio..... | 14 |
| 1. Create FASTA files and input files for BLAST / filtering..... | 15 |
| 2. Download genomes from the NCBI..... | 17 |
| 3. Create BLAST+ database..... | 19 |
| 4. BLAST and filtering..... | 21 |
| 5. Running BLAST and filtering script without the file with Unique ID..... | 24 |
| 6. Additional filtering..... | 25 |
| 7. Common errors..... | 26 |
| <i>omicR for Linux</i> | 27 |
| <i>omicR for HPC computers</i> | 28 |
| 1. Create FASTA files and input files for BLAST / filtering..... | 28 |
| 2. Download genomes from the NCBI..... | 29 |
| 3. Create NCBI database for BLAST+ | 31 |
| 4. BLAST and filtering..... | 31 |
| 5. Additional filtering..... | 34 |

Figures

| | |
|--|----|
| FIGURE 1 SAMPLE DATA OF E. FAECIUM | 6 |
| FIGURE 2 FIRST WINDOWS OF OMICR GUI..... | 6 |
| FIGURE 3 CREATE FASTA FILES AND INPUT FILES FOR BLAST USING GUI. | 7 |
| FIGURE 4 DISPLAY OF HOW ROWS AND HEADERS SHOULD BE SELECTED..... | 7 |
| FIGURE 5 OUTPUT FILES GENERATED FROM SCRIPT "CREATE FASTA FILES"..... | 8 |
| FIGURE 6 DOWNLOAD GENOME ENTRIES FROM NCBI..... | 8 |
| FIGURE 7 CREATE NCBI GENOME DATABASE FOR BLAST+..... | 9 |
| FIGURE 8 EXAMPLE OF FILES CREATED AFTER CREATING BLAST DATABASE..... | 9 |
| FIGURE 9 BLAST AND FILTERING EXAMPLE..... | 10 |
| FIGURE 10 EXAMPLE OF FILES PRODUCED AFTER RUNNING THE BLAST AND FILTERING SCRIPT..... | 10 |
| FIGURE 11 ADDITIONAL FILTERING WINDOW. | 12 |
| FIGURE 12 EXAMPLE OF CSV INPUT FILE..... | 14 |
| FIGURE 13 MKFASTAFILE.R SCRIPT BEFORE ADDING DETAILS..... | 15 |
| FIGURE 14 MKFASTAFILE.R SCRIPT AFTER ADDING DETAILS. | 16 |
| FIGURE 15 MKFASTAFILE.R SCRIPT AFTER SELECTING ALL THE CODE. | 16 |
| FIGURE 16 MKFASTAFILE.R SCRIPT AFTER RUNNING THE CODE..... | 17 |
| FIGURE 17 EXAMPLE OF FASTA FILE AND FILE WITH UNIQUE ID. | 17 |
| FIGURE 18 EXAMPLE OF DONWNLOADGENOMES.R SCRIPT..... | 18 |
| FIGURE 19 EXAMPLE OF DONWNLOADGENOMES.R SCRIPT AFTER ADDING PARAMETERS..... | 18 |
| FIGURE 20 EXAMPLE OF DONWNLOADGENOMES.R SCRIPT AFTER RUNNING THE CODE. | 19 |
| FIGURE 21 EXAMPLE OF WINDOW SHOWING "MAKEDATABASE.R" | 19 |
| FIGURE 22 EXAMPLE OF SCRIPT "MAKEDATABASE.R" AFTER FILLING PARAMETERS..... | 20 |
| FIGURE 23 EXAMPLE OF SCRIPT "MAKEDATABASE.R" AFTER RUNNING THE CODE. | 20 |
| FIGURE 24 EXAMPLE OF FILES CREATED WITH THE CODE "MAKEDATABASE.R". | 21 |
| FIGURE 25 EXAMPLE OF WINDOW SHOWING BLASTnFILTER.R SCRIPT..... | 21 |
| FIGURE 26 EXAMPLE OF SCRIPT BLASTnFILTER.R AFTER FILLING PARAMETERS..... | 22 |
| FIGURE 27 EXAMPLE OF SCRIPT BLASTnFILTER.R AFTER RUNNING THE CODE..... | 22 |
| FIGURE 28 EXAMPLE OF FILES CREATED BY BLASTnFILTER.R SCRIPT..... | 22 |
| FIGURE 29 EXAMPLE OF SCRIPT BLASTnFILTER.R AFTER RUNNING THE CODE WITHOUT THE FILE WITH UNIQUEID..... | 24 |
| FIGURE 30 EXAMPLE OUTPUT FILES GENERATED AFTER RUNNING THE SCRIPT BLASTnFILTER.R WITHOUT THE FILE WITH UNIQUEID.24 | 24 |
| FIGURE 31 EXAMPLE OF FILTER.R SCRIPT. | 25 |
| FIGURE 32 EXAMPLE OF "FILTER.R SCRIPT" AFTER FILLING PARAMETERS..... | 25 |
| FIGURE 33 EXAMPLE OF "FILTER.R SCRIPT" AFTER RUNNING THE CODE. | 26 |
| FIGURE 34 DISPLAY OF HOW ROWS AND HEADERS SHOULD BE SELECTED. | 29 |

Introduction

omicR creates fasta files, downloads genomes from NCBI using the refseq number, creates databases to run BLAST+, runs BLAST+ and filters these results to obtain the best match per sequence.

These scripts can be used to run BLAST alignment of short-read (DArTseq data) and long-read sequences (Illumina, PacBio... etc). You can use reference genomes from NCBI, genomes from your private collection, contigs, scaffolds or any other genetic sequence that you would like to use as reference.

You can skip this tutorial and watch the tutorial video in YouTube

Installing omicR on Windows (1:30 min)

<https://youtu.be/19zn7WoKtbg>

Using omicR with Graphical User Interface (~20 min)

<https://youtu.be/pdMio2vj-FM>

Using omicR in R Studio (~20 min)

<https://youtu.be/g46a3Gd1q6g>

Installing omicR with graphical user interface

Windows

Requirements:

BLAST+ latest version: <https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/>

Installation:

There are 2 options to run omicR in windows.

1) **Option 1.**

Download the code from GitHub:

https://github.com/BTalamantesBecerra/omicR_for_Windows.

Download the zip directory “omicR.zip”. Unzip this directory and double click the executable file “omicR.exe” with the image of the green parrot. This will open a window where you can run the scripts. You do not need to install anything as everything is compiled into this file and you can start running your analysis. You can see this step in this video: <https://youtu.be/19zn7WoKtbg>

2) **Option2.** If you cannot open the executable file, you may need to run the script directly through Python. For this you need to install the following:

Download the code from GitHub:

<https://github.com/BTalamantesBecerra/omicR>.

- a. Python V3 or latest: <https://www.python.org/downloads/>
- b. Biopython <https://biopython.org/>

-Open the script “omicR.py” in Python and run it.

-This will open a Window where you can run the scripts.

Linux

If you are using Linux, it is likely that Python is already installed. Download the code from GitHub: <https://github.com/BTalamantesBecerra/omicR>. For Linux, download the code from https://github.com/BTalamantesBecerra/omicR_for_linux

Requirements:

- a. Python V3 or latest: <https://www.python.org/downloads/>
 - i. Module: tkinter
- b. Biopython <https://biopython.org/>
 - i. Module: entrez
- c. BLAST+ latest version: <https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/>

-Open the script “omicR.py” in Python and run it.

-This will open a window where you can run the scripts.

Remember to add Python, Biopython and BLAST into your System Variables Path. As general practice, avoid installing your software in directories such as “C:\\Program files\\” as the space between words will cause problems.

Running “omicR” with graphical user interface

Before you run a test, download the Sample data. The csv file is called “**SampleData_Enterococcus_faecium.csv**”. **Only CSV files are accepted as input for these scripts.**

| SeqIndex | ClusterIdx | ClusterSize | Tag | TrimmedLength | LowComplexityChrom | ChromPos | AlnCnt | AlnValue | AdapterPc | NumPrese | HighestCo | CountSum | AvgNonZero | E | H | D | | |
|----------|------------|-------------|----------------------|---------------|--------------------|----------|--------|----------|-----------|----------|-----------|----------|------------|-------|--------------|--------------|--------|--|
| 1 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | E6 | H7 | D3 | |
| 2 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | 900716298002 | 900716298002 | 90 | |
| 3 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | 2 | 2 | | |
| 4 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | 152_2014_ENT | 163_2015_ENT | 12 | |
| 5 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | 161348 | | 151821 | |
| 6 | 89184 | 20963 | 408 TGCAGAA TGCAAGA | 49 | 0 | 0 | 0 | 999 | 49 | 43 | 51484 | 596359 | 13868.81 | 16911 | | 999079 | | |
| 7 | 75556 | 20878 | 111 TGCAGCG TGCAAGC | 46 | 0 | 0 | 0 | 999 | 46 | 14 | 12437 | 107456 | 7675.429 | 0 | | | 8728 | |
| 8 | 73901 | 20862 | 96 TGCACTA TGCACTA | 41 | 0 | 0 | 0 | 999 | 41 | 16 | 12381 | 90809 | 5675.563 | 0 | | | 6452 | |
| 9 | 79622 | 20911 | 137 TGCAGTT TGCAAGT | 40 | 0 | 0 | 0 | 999 | 40 | 82 | 3987 | 215378 | 2626.561 | 3001 | | | 2180 | |
| 10 | 78436 | 20902 | 127 TGCAGCT TGCAAGCT | 43 | 0 | 0 | 0 | 999 | 43 | 82 | 3163 | 180355 | 2199.451 | 2475 | | | 2096 | |
| 11 | 78175 | 20900 | 126 TGCACTA TGCACTA | 46 | 0 | 0 | 0 | 999 | 46 | 80 | 3093 | 162945 | 2036.813 | 2461 | | | 2068 | |
| 12 | 79343 | 20909 | 134 TGCAGAT TGCAAGAT | 52 | 0 | 0 | 0 | 999 | 52 | 81 | 2969 | 138818 | 1713.802 | 1827 | | | 1779 | |
| 13 | 76464 | 20886 | 118 TGCAAGA TGCAAGA | 43 | 0 | 0 | 0 | 999 | 43 | 83 | 4764 | 152191 | 1833.627 | 1943 | | | 1608 | |
| 14 | 79759 | 20912 | 137 TGCAGTTTGCAAGTT | 52 | 0 | 0 | 0 | 999 | 52 | 79 | 2560 | 132856 | 1681.722 | 1792 | | | 2310 | |
| 15 | 80025 | 20914 | 139 TGCAAGA TGCAAGA | 53 | 0 | 0 | 0 | 999 | 53 | 79 | 2736 | 133532 | 1690.278 | 2040 | | | 1657 | |
| 16 | 79492 | 20910 | 136 TGCACTA TGCACTA | 45 | 0 | 0 | 0 | 999 | 45 | 80 | 2759 | 138077 | 1725.963 | 2274 | | | 1664 | |
| 17 | 80161 | 20915 | 141 TGCAGAG TGCAAG | 51 | 0 | 0 | 0 | 999 | 51 | 79 | 3700 | 133197 | 1686.038 | 1728 | | | 1621 | |
| 18 | 76107 | 20883 | 117 TGCAGTA TGCAAGTA | 46 | 0 | 0 | 0 | 999 | 46 | 80 | 2686 | 131079 | 1638.488 | 1814 | | | 1622 | |
| 19 | 82781 | 20933 | 156 TGCAAGTTTGCAAGTT | 69 | 0 | 0 | 0 | 999 | 0 | 80 | 2355 | 114989 | 1437.363 | 1580 | | | 1466 | |

Figure 1 Sample Data of *E. faecium*

Open or run the “omicR” executable. The window should look like this. As you can see, each button runs a different script.

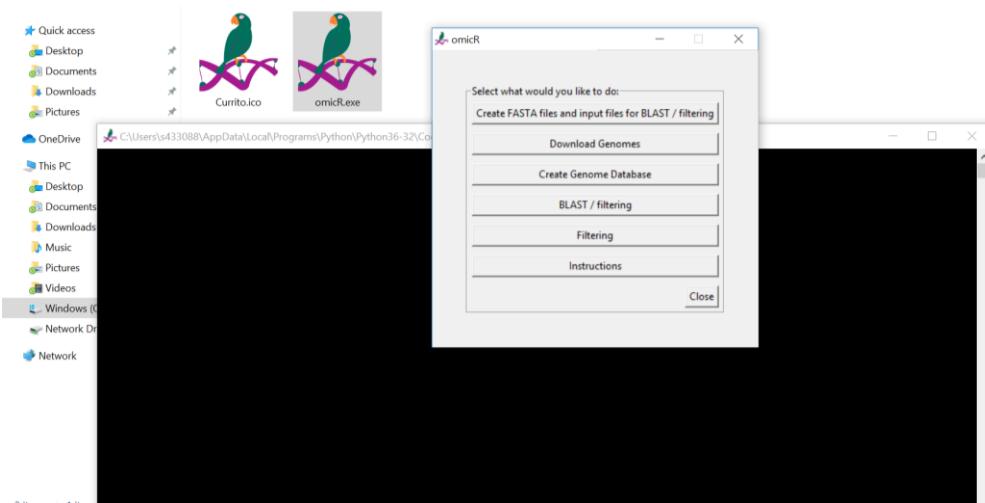


Figure 2 First windows of omicR GUI.

STEPS

1. Create FASTA files and input files for BLAST / filtering.

This will open another window. Select the paths to the Sample Data file provided and select parameters as required.

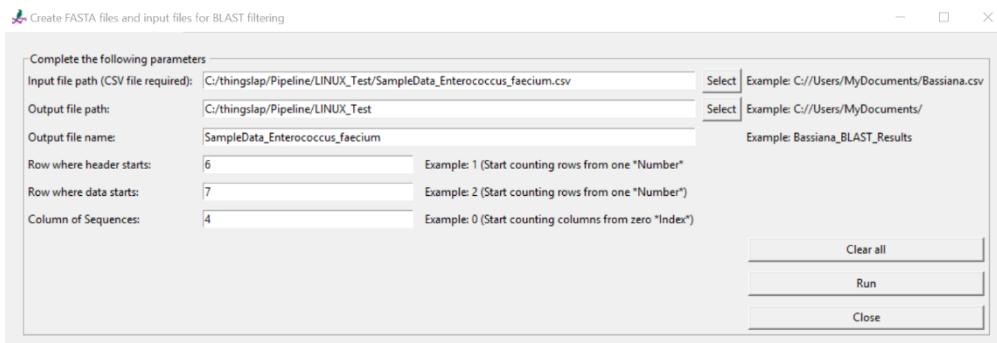


Figure 3 Create Fasta files and input files for BLAST using GUI.

The row where header starts is row 6 and the row where data starts is 7. And in this case, for this file we need the Trimmed Sequences column, it is in column 4. For columns you need to start counting from 0.

Sequence of interest: Column 4

| | A | B | C | D | E | F | G | H | I | J | K |
|----|----------|------------|-------------|---------|----------------|--------------|------------|--------|------------|-----|---|
| 1 | * | * | * | * | * | * | * | * | * | * | * |
| 2 | * | * | * | * | * | * | * | * | * | * | * |
| 3 | * | * | * | * | * | * | * | * | * | * | * |
| 4 | * | * | * | * | * | * | * | * | * | * | * |
| 5 | * | * | * | * | * | * | * | * | * | * | * |
| 6 | SeqIndex | ClusterIdx | ClusterSize | Tag | TrimmedSLength | LowComlChrom | deChromPos | AlnCnt | dsAlnValue | | |
| 7 | 89184 | 20963 | 408 | TGCAGAA | TGCAGAA | 49 | 0 | 0 | 0 | 999 | |
| 8 | 75556 | 20878 | 111 | TGCAGCG | TGCAGCG | 46 | 0 | 0 | 0 | 999 | |
| 9 | 73901 | 20862 | 96 | TGCAGTA | TGCAGTA | 41 | 0 | 0 | 0 | 999 | |
| 10 | 79622 | 20911 | 137 | TGCAGTT | TGCAGTT | 40 | 0 | 0 | 0 | 999 | |
| 11 | 78436 | 20902 | 127 | TGCAGCT | TGCAGCT | 43 | 0 | 0 | 0 | 999 | |
| 12 | 78175 | 20900 | 126 | TGCAGCA | TGCAGCA | 46 | 0 | 0 | 0 | 999 | |
| 13 | 79343 | 20909 | 134 | TGCAGAT | TGCAGAT | 52 | 0 | 0 | 0 | 999 | |

Figure 4 Display of how rows and headers should be selected.

REMEMBER TO CHECK THE HEADER AND DATA OF YOUR FILE BEFORE RUNNING ANY SCRIPT

This will create 2 files, one Fasta file and one copy of your original file including an extra column containing a Unique ID. These files are required for the following steps.

Fasta file

| |
|--|
| 1 >1 |
| 2 TGCAGAAGAAGTACGAAGAGAACAGAACTCTACGCCGAAACACCG |
| 3 >2 |
| 4 TGCAGGGCCATCATACGGGATAACGACTGTATGACGTGAAACCG |
| 5 >3 |
| 6 TGCAGTACGGAATCTTCGATTATCAGGAAGTCGAGCCG |
| 7 >4 |
| 8 TGCAGTTGCTGTTCTGGCACCATTTTGCAGTCGAGTCGG |
| 9 >5 |
| 10 TGCAGCTGCATTGGCTCGATTACTTGATGCAAGAACATCCG |
| 11 >6 |
| 12 TGCAGCATCGCTTGAAGAACTAGGCGTTACGTATTAGAACCG |
| 13 >7 |
| 14 TGCAGATGATACCGTTACCTAGCTGAACGCATTAGAGAACAAACTACCG |
| 15 >8 |
| 16 TGCAGAACGCATCATATATTGGCTTAACGATTGTGCCCC |
| 17 >9 |
| 18 TGCAGTTCTGGTAAATTCTCTAGCATCAACCAAGAACATCGTACCG |
| 19 >10 |
| 20 TGCAGGAGCTGTTTGAGTTACAGAACCGAGAACGCTACAAACCG |

File with FastaFileID

| A | B | C | D | E | TrimmedSequence |
|----|-------------|----------|------------|-------------|---|
| 1 | FastaFileID | SeqIndex | ClusterIdx | ClusterSize | Tag |
| 2 | 1 | 89184 | 20963 | 408 | TGCAGAAGAAGTACGAAGAGAACAGAACTCTACGCCGAAACACCG |
| 3 | 2 | 75556 | 20878 | 111 | TGCAGCG TGCGAGGGCCATCATACGGGAT |
| 4 | 3 | 73901 | 20862 | 96 | TGCAGTA(TGCAGTACGGAACTCTTCGATT) |
| 5 | 4 | 79622 | 20911 | 137 | TGCAGTT(TGCAGTTGCTTCCGGACCAT |
| 6 | 5 | 78436 | 20902 | 127 | TGCAGCT(TGCAGCTGCTATTGGCTTCGATTACG) |
| 7 | 6 | 78175 | 20900 | 126 | TGCAGCA(TGCAGCATCGCTTGAAGAACATCG) |
| 8 | 7 | 79343 | 20909 | 134 | TGCAGAT(TGCAGATGATATCGTTACCTAGG) |
| 9 | 8 | 76464 | 20886 | 118 | TGCAGAA(TGCAGAACGCATCATATATTCG) |
| 10 | 9 | 79759 | 20912 | 137 | TGCAGTTT(TGCAGTTTCTGGTAATTTCCTC) |
| 11 | 10 | 80025 | 20914 | 139 | TGCAGGA TGCGAGGAGCTGTTTGAGTTACG |
| 12 | 11 | 79492 | 20910 | 136 | TGCAGCA(TGCAGCTGGCTGTTGACTTACG) |
| 13 | 12 | 80161 | 20915 | 141 | TGCAGAG TGCAGAGAAACTCGATCCACTG |
| 14 | 13 | 76107 | 20883 | 117 | TGCAGTA(TGCAGTATTAAATGACAAATGCG) |
| 15 | 14 | 82781 | 20933 | 156 | TGCAGTT(TGCAGTTGCAAGTGCAGAACAGTC) |
| 16 | 15 | 75759 | 20880 | 112 | TGCAGCA(TGCAGCAAAGTACTAGTAGAGAA) |
| 17 | 16 | 85903 | 20950 | 214 | TGCAGTT(TGCAGTTGAGGAAAAGAAAAAAAG) |
| 18 | 17 | 83279 | 20936 | 163 | TGCAGTA(TGCAGTAGCCATTACGTGACCG) |
| 19 | 18 | 83781 | 20939 | 179 | TGCAGCA(TGCAGCACACGGATACCGTCTT) |
| 20 | 19 | 83415 | 20937 | 171 | TGCAGAT(TGCAGATCAAATGTAGAAAGATA) |

Figure 5 Output files generated from script "Create Fasta files".

2. Download genomes from the NCBI.

For this example, we need the genome and plasmid of *E. faecium* (https://www.ncbi.nlm.nih.gov/assembly/GCF_010120755.1). The RefSeq numbers needed for this tutorial are: NZ_CP039729.1, NZ_CP039730.1

To run the script, write your email, select the RefSeq accession numbers, select the output path and name for the genome.

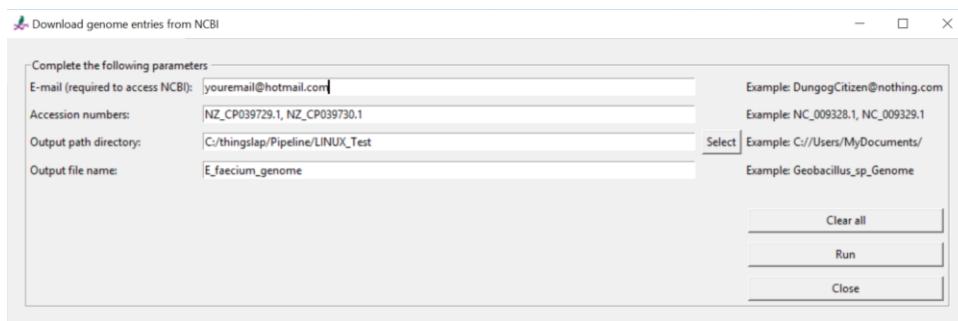


Figure 6 Download genome entries from NCBI.

This will create a directory with the name given to the genome.

Please note that this option is suitable for small genomes or small chromosomes. To download larger genomes, it is recommended to use the internet browser option. This example of *E. faecium* should download in less than 5 minutes. If you use this method to download the Chicken genome it can take up to 5 hours.

3. Create NCBI database for BLAST+.

To create the database for BLAST, select the path to the “bin” directory where BLAST+ was installed.

Select the location of the genome/sequences/scaffolds/contigs or whatever you would like to use as reference to BLAST, then select the type of database. For this example we are working with nucleotides, so we select “nucl”.

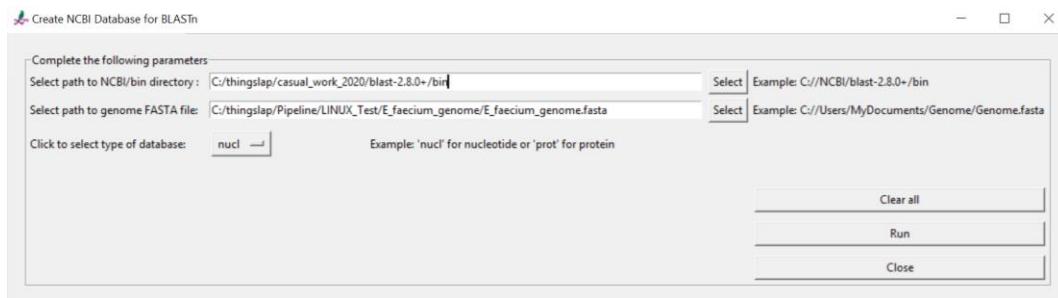


Figure 7 Create NCBI genome database for BLAST+.

The script will create 3 files in the same location as your database, with the same name of the reference and the terminations “.nhr”, “.nin”, “.nsq”.

| Name | Date modified | Type | Size |
|----------------------------|---------------------|------------|----------|
| E_faecium_genome.fasta | 25/08/2020 11:57 AM | FASTA File | 2,769 KB |
| E_faecium_genome.fasta.nhr | 25/08/2020 12:08 PM | NHR File | 1 KB |
| E_faecium_genome.fasta.nin | 25/08/2020 12:08 PM | NIN File | 1 KB |
| E_faecium_genome.fasta.nsq | 25/08/2020 12:08 PM | NSQ File | 692 KB |

Figure 8 Example of files created after creating BLAST database.

4. BLAST and filtering.

To run the BLAST analysis, you need to select the path to the bin directory where you installed BLAST+, then select your path to the database created in the previous step, the output path, the output file name, the output path to the file with Unique ID and BLAST parameters.

NOTES:

- If you only used a fasta file as input, and you do not have the file with Unique ID, you can still run this script.
- If you are running a BLAST alignment of similar sequences, for example Turtle Genome Vs Turtle Sequences, the recommended parameters are: Word Size 11, Percentage identity 70, Number of threads 4, Output format 6, Percentage Overlap 0.8, bitscore 50.
- If you are running a BLAST of highly dissimilar sequences because you are probably looking for sex linked hits in a distantly related species, and you are aligning sequences of Chicken Genome Vs Bassiana, use a Percentage overlap of 0.01, Bitscore of 30 and tick the option of “Discontinuous Mega BLAST”

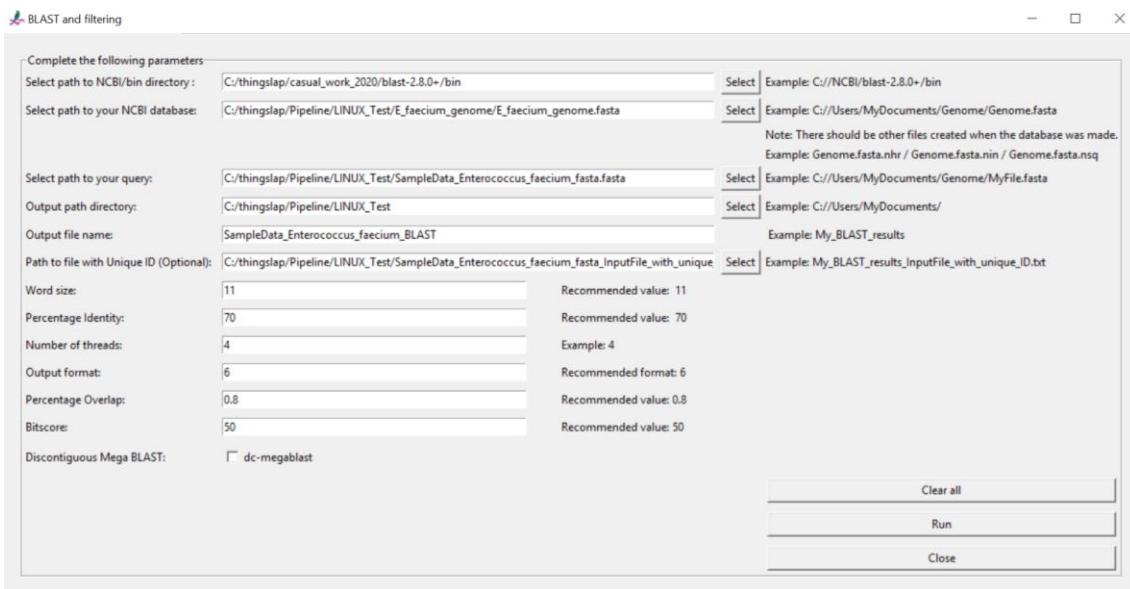


Figure 9 BLAST and filtering example.

This step takes less than 5 minutes. It will produce 5 files, or 3 files if you didn't provide the File with UniqueID.

| | | | |
|---|--------------------|---------------|----------|
| SampleData_Enterococcus_faecium_BLAST_all_sequences_with_and_without_hits.txt | 24/08/2020 5:53 PM | Text Document | 425 KB |
| SampleData_Enterococcus_faecium_BLAST_only_sequences_with_hits.txt | 24/08/2020 5:53 PM | Text Document | 377 KB |
| SampleData_Enterococcus_faecium_BLAST_sorted.txt | 24/08/2020 5:52 PM | Text Document | 215 KB |
| SampleData_Enterococcus_faecium_BLAST_filtered.txt | 24/08/2020 5:52 PM | Text Document | 257 KB |
| SampleData_Enterococcus_faecium_BLASTBLAST.txt | 24/08/2020 5:52 PM | Text Document | 3,057 KB |

Figure 10 Example of files produced after running the BLAST and filtering script.

Description of files produced.

- **File 1. SampleData_Enterococcus_faecium_BLASTBLAST.txt**

This is the raw BLAST output. This file does not contain any headers and it is not filtered.

- **File 2. SampleData_Enterococcus_faecium_BLAST_filtered.txt**

This file has headers and an extra column with the percentage overlap. Default filtering parameters for this tutorial are: Percentage Overlap >80%, bitscore >50, Percentage Identity>70. The percentage overlap can be modified according to the BLAST results expected. This file may contain multiple hits per sequence.

The BLAST output is formatted as a table using output format 6, with columns defined in the following order: " qseqid sacc stitle qseq sseq nident mismatch pident length evalue bitscore qstart qend sstart send gapopen gaps qlen slen". These are:

- qseqid: query (e.g., unknown gene) sequence id
- sacc: Subject accession
- stitle: Subject Title
- qseq: Aligned part of query sequence
- sseq: Aligned part of subject sequence
- nident: Number of identical matches
- mismatch: number of mismatches
- pident: percentage of identical matches
- length: alignment length (sequence overlap)
- evalue: expect value
- bitscore: bit score
- qstart: start of alignment in query
- qend: end of alignment in query
- sstart: start of alignment in subject
- send: end of alignment in subject
- gapopen: number of gap openings
- gaps: Total number of gaps
- qlen: Query sequence length
- slen: Subject sequence length

- **File 3. SampleData_Enterococcus_faecium_BLAST_sorted.txt**

This file contains only one hit per sequence. The best match will be selected by considering the following values ranked in order. First considering the highest percentage identity, then the highest Percentage overlap, then the highest bitscore. Only one Query per sequence is kept based on these selection criteria.

If you did not provide the file with UniqueID, this filtered and sorted BLAST output file will be your final result.

- **File 4. SampleData_Enterococcus_faecium_BLAST_only_sequences_with_hits.txt**

This file uses the UniqueID assigned to each sequence and writes the BLAST results back into the original file. This file only contains sequences that had a BLAST hit to something in the reference.

- **File 5. SampleData_Enterococcus_faecium_BLAST_all_sequences_with_and_without_hits.txt**

This file uses the UniqueID assigned to each sequence and writes the BLAST results back into the original file. This file contains all sequences, including those with and without hits, written back into the original file.

5. Additional filtering.

If you would like to run additional filtering without re-running the BLAST, you can use the BLAST result obtained in the previous step as an input and filter again with different parameters. For example, using a higher or lower percentage overlap or bitscore.

Note: This script only takes input files with BLAST Tabular output format 6 with the ordered set of columns described in the previous step.

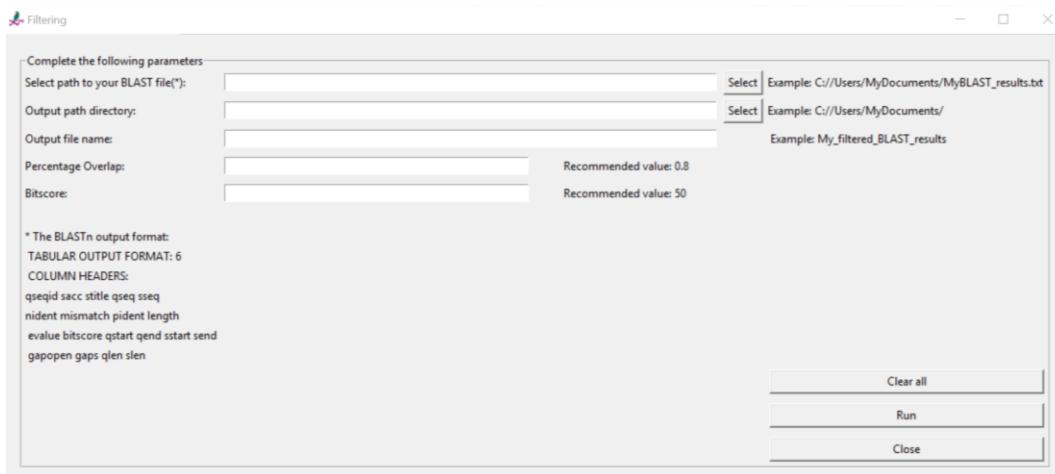


Figure 11 Additional filtering window.

omicR for R Studio

omicR for R studio runs using Python scripts through R Studio. You can download the R project in:
https://github.com/BTalamantesBecerra/omicR_for_RStudio

- 1) You need to install the following:
 - a. Python V3 or latest: <https://www.python.org/downloads/>
 - b. Biopython <https://biopython.org/>
 - c. Download the omicR code. This should include at least the following:
“omicR.Rproj”,
“TestingPyCharm_MKfasta.py”,
“TestingPyCharm_Downloading_genomes.py”,
“TestingPyCharm_MakeDataBase.py”,
“TestingPyCharm_BLAST_filtering_and_all.py”,
“TestingPyCharm_NCBI_BLAST_filtering.py”,

Running “omicR” in R studio

Before you run a test, download the Sample data. The csv file is called “SampleData_Enterococcus_faecium.csv”.

Note: The pipeline only takes csv files as input.

| SeqIndex | ClusterIdx | ClusterSize | Tag | TrimmedSLength | LowComlChrom | ChromPos | AlnCnt_da | AlnEval | AdapterPc | NumPres | HighestCo | CountSum | AvgNonZe | E6 | H7 | D3 | |
|----------|------------|-------------|-----|----------------|--------------|----------|-----------|---------|-----------|---------|-----------|----------|----------|----------------------------|--------------|------|--|
| 1 | * | * | * | * | * | * | * | * | * | * | * | * | * | * | | | |
| 2 | * | * | * | * | * | * | * | * | * | * | * | * | * | 900716298002 | 900716298002 | 90 | |
| 3 | * | * | * | * | * | * | * | * | * | * | * | * | * | 2 | 2 | | |
| 4 | * | * | * | * | * | * | * | * | * | * | * | * | * | 152_2014_ENT163_2015_ENT12 | | | |
| 5 | * | * | * | * | * | * | * | * | * | * | * | * | * | 161348 | 151821 | | |
| 6 | 89184 | 20963 | 408 | TGCAAGAA | TGCAAGAA | 49 | 0 | 0 | 999 | 49 | 43 | 51484 | 596359 | 13868.81 | 16911 | 0 | |
| 7 | 75556 | 20878 | 111 | TGCAGCG | TGCAGCG | 46 | 0 | 0 | 999 | 46 | 14 | 12437 | 107456 | 7675.429 | 0 | 8728 | |
| 8 | 73901 | 20862 | 96 | TGCACTA | TGCACTA | 41 | 0 | 0 | 999 | 41 | 16 | 12381 | 90809 | 5675.563 | 0 | 6452 | |
| 9 | 79622 | 20911 | 137 | TGCACTT | TGCACTT | 40 | 0 | 0 | 999 | 40 | 82 | 3987 | 215378 | 2626.561 | 3001 | 2180 | |
| 10 | 78436 | 20902 | 127 | TGCACTC | TGCACTC | 43 | 0 | 0 | 999 | 43 | 82 | 3163 | 180355 | 2199.451 | 2475 | 2096 | |
| 11 | 78175 | 20900 | 126 | TGCACTA | TGCACTA | 46 | 0 | 0 | 999 | 46 | 80 | 3093 | 162945 | 2036.813 | 2461 | 2068 | |
| 12 | 79343 | 20909 | 134 | TGCACTT | TGCACTT | 52 | 0 | 0 | 999 | 52 | 81 | 2969 | 138818 | 1713.802 | 1827 | 1779 | |
| 13 | 76464 | 20886 | 118 | TGCAAGAA | TGCAAGAA | 43 | 0 | 0 | 999 | 43 | 83 | 4764 | 152191 | 1833.627 | 1943 | 1608 | |
| 14 | 79759 | 20912 | 137 | TGCACTTT | TGCACTTT | 52 | 0 | 0 | 999 | 52 | 79 | 2560 | 132856 | 1681.722 | 1792 | 2310 | |
| 15 | 80025 | 20914 | 139 | TGCAAGA | TGCAAGA | 53 | 0 | 0 | 999 | 53 | 79 | 2736 | 133532 | 1690.278 | 2040 | 1657 | |
| 16 | 79492 | 20910 | 136 | TGCACTA | TGCACTA | 45 | 0 | 0 | 999 | 45 | 80 | 2759 | 138077 | 1725.963 | 2274 | 1664 | |
| 17 | 80161 | 20915 | 141 | TGCACTG | TGCACTG | 51 | 0 | 0 | 999 | 51 | 79 | 3700 | 133197 | 1686.038 | 1728 | 1621 | |
| 18 | 76107 | 20883 | 117 | TGCACTA | TGCACTA | 46 | 0 | 0 | 999 | 46 | 80 | 2686 | 131079 | 1638.488 | 1814 | 1622 | |
| 19 | 82781 | 20933 | 156 | TGCACTTT | TGCACTTT | 69 | 0 | 0 | 999 | 0 | 80 | 2335 | 114989 | 1437.363 | 1580 | 1466 | |

Figure 12 Example of CSV input file.

STEPS

1. Create FASTA files and input files for BLAST / filtering.

- Select the script “mkfastafolder.R”.
 - Clean the global environment before running these scripts.

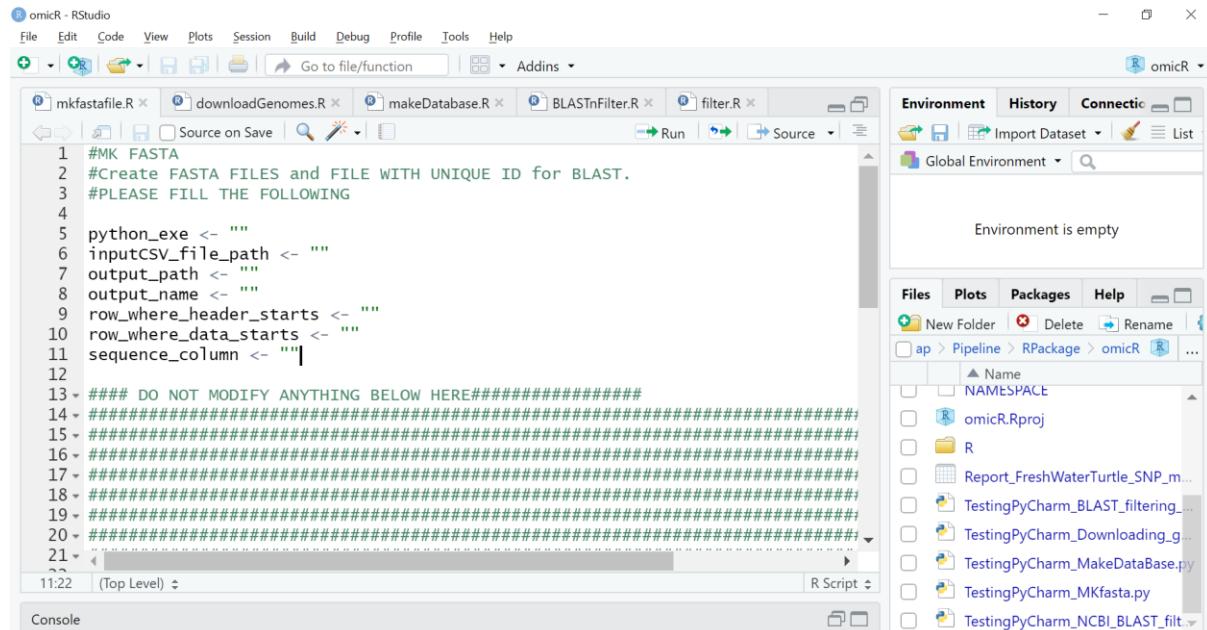


Figure 13 mkfastafile.R script before adding details.

- Now write the paths to your files as indicated. Do not modify anything below the sign “DO NOT MODIFY ANYTHING BELOW HERE”.

Notes:

- Remember to use double backslashes “\\” to allow the script to run successfully.
 - After you fill the parameters, it should look like this:

The screenshot shows the RStudio interface with the 'omicR - RStudio' window open. The left pane displays the R script 'mkfastafile.R' containing the following code:

```

1 #MK FASTA
2 #Create FASTA FILES and FILE WITH UNIQUE ID for BLAST.
3 #PLEASE FILL THE FOLLOWING
4
5 python_exe <- "C:\\Users\\s433088\\AppData\\Local\\Programs\\Python\\Python38\\python.exe"
6 inputCSV_file_path <- "C:\\thingslap\\Pipeline\\LINUX_Test\\SampleData_Enterococcus_faecium.csv"
7 output_path <- "C:\\thingslap\\Pipeline\\LINUX_Test\\"
8 output_name <- "SampleData_Enterococcus_faecium_R"
9 row_where_header_starts <- "6"
10 row_where_data_starts <- "7"
11 sequence_column <- "4"
12
13 ##### DO NOT MODIFY ANYTHING BELOW HERE#####
14 #####
15 #####
16 #####
17 #####
18 #####
19 #####
20 #####
21 #####

```

The right pane shows the environment and files sections. The environment section says 'Environment is empty'. The files section shows a folder structure with 'New Folder', 'R...', 'Untitled.R...', and 'Untitled.R'.

Figure 14 mkfastafile.R script after adding details.

- Press CTRL+A to select the entire code

The screenshot shows the RStudio interface with the 'omicR - RStudio' window open. The left pane displays the R script 'mkfastafile.R' with the entire code highlighted in blue, indicating it is selected.

```

1 #MK FASTA
2 #Create FASTA FILES and FILE WITH UNIQUE ID for BLAST.
3 #PLEASE FILL THE FOLLOWING
4
5 python_exe <- "C:\\Users\\s433088\\AppData\\Local\\Programs\\Python\\Python38\\python.exe"
6 inputCSV_file_path <- "C:\\thingslap\\Pipeline\\LINUX_Test\\SampleData_Enterococcus_faecium.csv"
7 output_path <- "C:\\thingslap\\Pipeline\\LINUX_Test\\"
8 output_name <- "SampleData_Enterococcus_faecium_R"
9 row_where_header_starts <- "6"
10 row_where_data_starts <- "7"
11 sequence_column <- "4"
12
13 ##### DO NOT MODIFY ANYTHING BELOW HERE#####
14 #####
15 #####

```

Figure 15 mkfastafile.R script after selecting all the code.

- Run the code

```

1 #MK FASTA
2 #Create FASTA FILES and FILE WITH UNIQUE ID for BLAST.
3 #PLEASE FILL THE FOLLOWING
4
5 python_exe <- "C:\\Users\\s433088\\AppData\\Local\\Programs\\Python\\Python38\\"
6 inputCSV_file_path <- "C:\\thingslap\\Pipeline\\LINUX_Test\\SampleData_Enterococcus_faecium_R.csv"
7 output_path <- "C:\\thingslap\\Pipeline\\LINUX_Test\\"
8 output_name <- "SampleData_Enterococcus_faecium_R"
9 row_where_header_starts <- "6"
10 row_where_data_starts <- "7"
11 sequence_column <- "4"
12
13 ##### DO NOT MODIFY ANYTHING BELOW HERE#####
14 #####
15 1

```

(Top Level)

Console

```

C:/thingslap/Pipeline/RPackage/omicR/
+ "-f", sequence_column
+ )
> system(python_command_line_TestingPycharm_Mkfasta)
[1] 0
>

```

Figure 16 mkfastafile.R script after running the code.

This code will produce two files that are needed for the following steps.

| | A | B | C | D | E |
|----|--|----|-------|-------|--------------------------------------|
| 1 | >1 | 1 | 89184 | 20963 | 408 TGCAGAAAAGTACGAAGAGAA |
| 2 | TGCAGAGAAAGTACGAAGAGAACAGAACGAACTTTACGCCCTGAACAAACCG | | | | |
| 3 | >2 | 2 | 75556 | 20878 | 111 TGCAGCGGTGCAGGGCCATCATACGGGGAT |
| 4 | TGCAGCGGCCATCATACGGGGATAACGACTGTATGACGTGAAACCG | | | | |
| 5 | >3 | 3 | 73901 | 20862 | 96 TGCAGTA(TGCAGTACGGAACTTTCTGATT |
| 6 | TGCAGTACGGAAATCTTCGATTTCATCAGGAAGTCGAGCGC | | | | |
| 7 | >4 | 4 | 79622 | 20911 | 137 TGCAGTT(TGCAGTTCTGCTGGCTTGACCACT |
| 8 | TGCAGTTGCTGTTCTGGCACCATTTTCGCGAAGTCGG | | | | |
| 9 | >5 | 5 | 78436 | 20902 | 127 TGCAGCT(TGCAGCTGCTGGCTTGATTAG |
| 10 | TGCAGCTGCATTGGCTTCGATTACTTGATGCAAGAACATCCG | | | | |
| 11 | >6 | 6 | 78175 | 20900 | 126 TGCAGCA(TGCAGCATCGCTTGAAGAACTA |
| 12 | TGCAGCATCGCTTGAAAGAACTAGGCCTTACGTGATTAGAACCG | | | | |
| 13 | >7 | 7 | 79343 | 20909 | 134 TGCAGAT(TGCAGATGATATCCGTTACCTAG |
| 14 | TGCAGATGATATCCGTTACGTGAAACGCTTAGAGAACGAAACATCCG | | | | |
| 15 | >8 | 8 | 76464 | 20886 | 118 TGCAGAA(TGCAGAACGCTCATCATATT |
| 16 | TGCAGAACGCTCATATATTGGCTTAACGATTGTGCCCC | | | | |
| 17 | >9 | 9 | 79759 | 20912 | 137 TGCAGTT(TGCAGTTCTGGTAATTTCCTC |
| 18 | TGCAGTTCTGGTAAATTTCCTAGCATCACCCAAGAACGATCGTACCG | | | | |
| 19 | >10 | 10 | 80025 | 20914 | 139 TGCAGGA(TGCAGGAGCTTTTGAGTTAC |
| 20 | TGCAGGAGCTGTTTGTAGTTACAGAACCGAGAACGGAGAACGCTACAAACCG | | | | |

SampleData_Enterococcus_faecium

Figure 17 Example of fasta file and file with Unique ID.

2. Download genomes from the NCBI.

- Select the script “donwnloadGenomes.R”.
- Clean your global environment before running this script.

```

1 #DOWNLOAD GENOMES
2
3 #Download genomes from NCBI using accession number.
4 #PLEASE FILL THE FOLLOWING
5
6 python_exe <- ""
7 email <- ""
8 genomeAccessions <- ""
9 OutputFilePath <- ""
10 fileName <- ""
11
12
13 ##### DO NOT MODIFY ANYTHING BELOW HERE#####
14 #####
15 <-
16 (Top Level) <

```

Figure 18 Example of downloadGenomes.R script.

- Fill the scripts with the path to the python executable files, your email, the RefSeq numbers separated by a comma and WITHOUT SPACE IN BETWEEN, the output file path and name.
- Remember to use \\ in all your paths.

```

1 #DOWNLOAD GENOMES
2
3 #Download genomes from NCBI using accession number.
4 #PLEASE FILL THE FOLLOWING
5
6 python_exe <- "C:\\Users\\s433088\\AppData\\Local\\Programs\\Python\\Python38\\python.exe"
7 email <- "berenicetalamantes@yahoo.fr"
8 genomeAccessions <- "NZ_CP039729.1,NZ_CP039730.1"
9 OutputFilePath <- "C:\\thingslap\\Pipeline\\LINUX_Test\\"
10 fileName <- "E_faecium_genome"
11
12
13 ##### DO NOT MODIFY ANYTHING BELOW HERE#####
14 #####
15 <-
16 (Top Level) <

```

Figure 19 Example of downloadGenomes.R script after adding parameters.

- Save your script, select all the code and run.

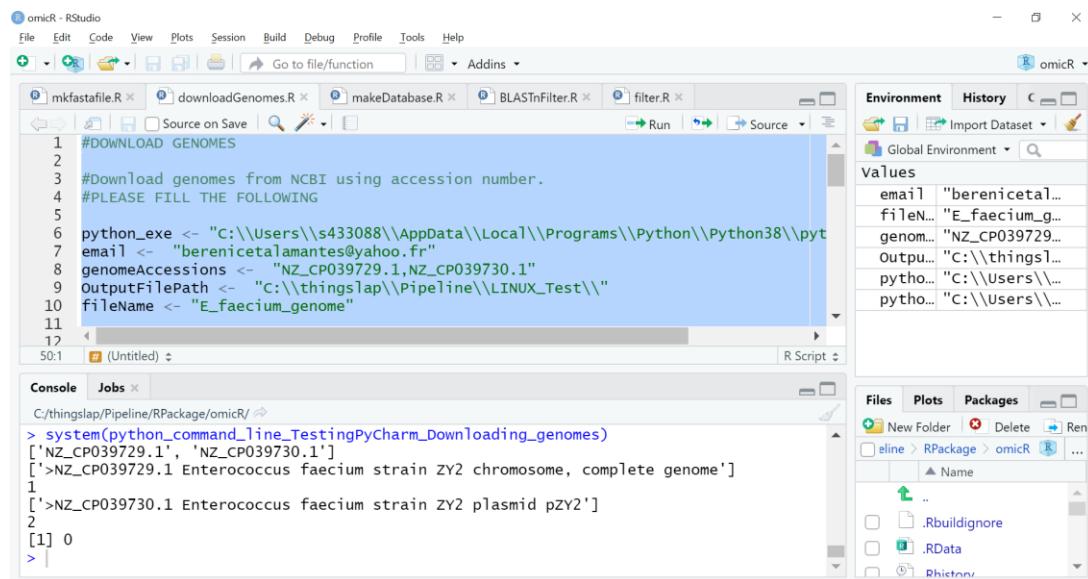


Figure 20 Example of downloadGenomes.R script after running the code.

This step creates a directory with a single fasta file that includes all the RefSeq numbers fetched.

3. Create BLAST+ database.

- Select the script “makeDatabase.R” and complete the script by typing the paths for your computer.
 - Clean your global environment before running this script.

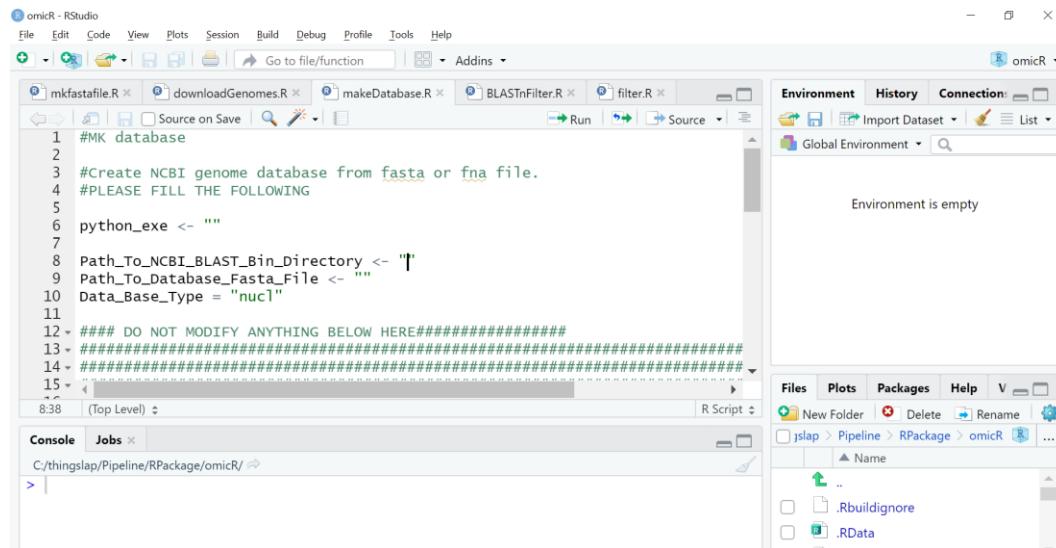
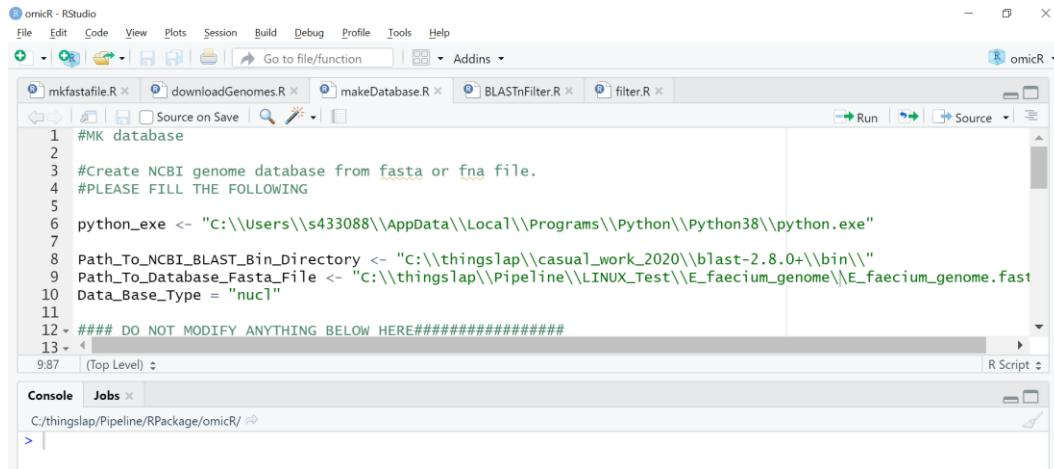


Figure 21 Example of window showing “makeDatabase.R”

- Fill the script and add the path to the Python executable, path to the BLAST+ bin directory, path to the file (genome, contigs, scaffolds...etc) to turn into database, and the type of database (for nucleotides select “nucl”).

- Remember to type \\ in the path.



The screenshot shows the RStudio interface with the 'makeDatabase.R' script open. The code defines variables for Python executable path, NCBI BLAST bin directory, database fasta file, and data base type. The 'Data_Base_Type' variable is set to 'nuc1'. A note at the bottom of the script indicates not to modify anything below it. The RStudio interface includes tabs for 'mkfastafolder.R', 'downloadGenomes.R', 'makeDatabase.R', 'BLASTnFilter.R', and 'filter.R'. Below the code editor is a 'Console' tab showing the command 'C:/thingslap/Pipeline/RPackage/omicR/ > |'.

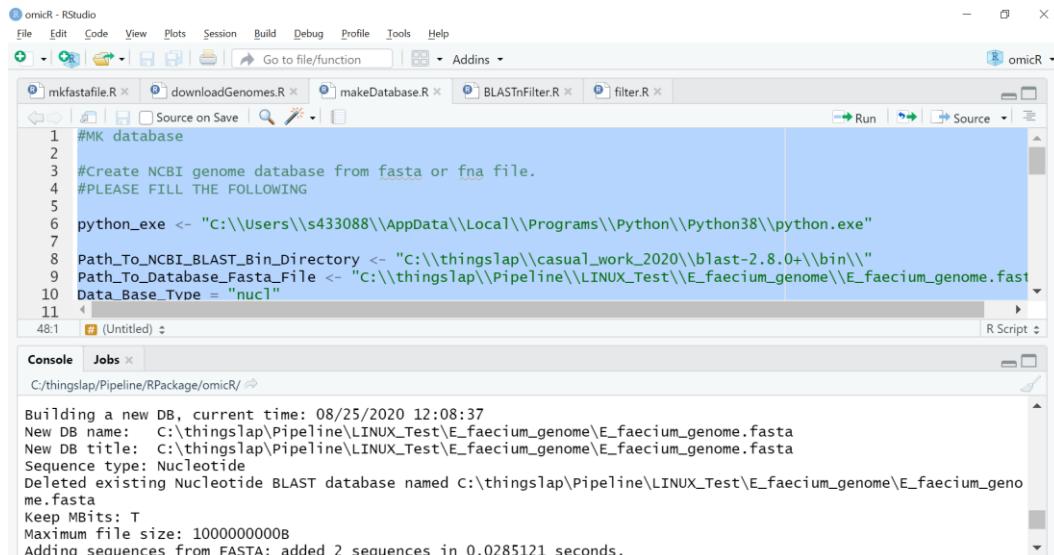
```

1 #MK database
2
3 #Create NCBI genome database from fasta or fna file.
4 #PLEASE FILL THE FOLLOWING
5
6 python_exe <- "C:\\\\users\\\\s433088\\\\AppData\\\\Local\\\\Programs\\\\Python\\\\Python38\\\\python.exe"
7
8 Path_To_NCBI_BLAST_Bin_Directory <- "C:\\\\thingslap\\\\casual_work_2020\\\\blast-2.8.0+\\\\bin\\\\"
9 Path_To_Database_Fasta_File <- "C:\\\\thingslap\\\\Pipeline\\\\LINUX_Test\\\\E_faecium_genome\\\\E_faecium_genome.fasta"
10 Data_Base_Type = "nuc1"
11
12 ##### DO NOT MODIFY ANYTHING BELOW HERE#####
13
9:87 (Top Level) ♦

```

Figure 22 Example of script “makeDatabase.R” after filling parameters.

- Save the script, select all the code and run it.



The screenshot shows the RStudio interface with the 'makeDatabase.R' script running. The console output shows the creation of a new BLAST database named 'E_faecium_genome'. The output includes the current time, new DB name, new DB title, sequence type (Nucleotide), and the process of deleting existing files and adding sequences from the FASTA file. The command 'Adding sequences from FASTA; added 2 sequences in 0.0285121 seconds.' is visible.

```

1 #MK database
2
3 #Create NCBI genome database from fasta or fna file.
4 #PLEASE FILL THE FOLLOWING
5
6 python_exe <- "C:\\\\users\\\\s433088\\\\AppData\\\\Local\\\\Programs\\\\Python\\\\Python38\\\\python.exe"
7
8 Path_To_NCBI_BLAST_Bin_Directory <- "C:\\\\thingslap\\\\casual_work_2020\\\\blast-2.8.0+\\\\bin\\\\"
9 Path_To_Database_Fasta_File <- "C:\\\\thingslap\\\\Pipeline\\\\LINUX_Test\\\\E_faecium_genome\\\\E_faecium_genome.fasta"
10 Data_Base_Type = "nuc1"
11
48:1 (Untitled) ♦

```

Building a new DB, current time: 08/25/2020 12:08:37
New DB name: C:\\thingslap\\Pipeline\\LINUX_Test\\E_faecium_genome\\E_faecium_genome.fasta
New DB title: C:\\thingslap\\Pipeline\\LINUX_Test\\E_faecium_genome\\E_faecium_genome.fasta
Sequence type: Nucleotide
Deleted existing Nucleotide BLAST database named C:\\thingslap\\Pipeline\\LINUX_Test\\E_faecium_genome\\E_faecium_genome.fasta
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 2 sequences in 0.0285121 seconds.

Figure 23 Example of script “makeDatabase.R” after running the code.

- This script produces 3 additional files with the same name as the file used to create the database. These files have the terminations: “nhr”, “nin” and “nsq”.

| Name | Date modified | Type | Size |
|----------------------------|---------------------|------------|----------|
| E_faecium_genome.fasta | 25/08/2020 11:57 AM | FASTA File | 2,769 KB |
| E_faecium_genome.fasta.nhr | 25/08/2020 12:08 PM | NHR File | 1 KB |
| E_faecium_genome.fasta.nin | 25/08/2020 12:08 PM | NIN File | 1 KB |
| E_faecium_genome.fasta.nsq | 25/08/2020 12:08 PM | NSQ File | 692 KB |

Figure 24 Example of files created with the code “makeDatabase.R”.

4. BLAST and filtering.

- Select the script “BLASTnFilter.R”.
- Clean your global environment before running this script.

```

1 #This script BLAST nucleotides sequences and filters
2 #
3 #PLEASE FILL THE FOLLOWING
4
5 python_exe <- ""
6 Path_to_NCBI_Directory <- ""
7 DC_MegaBlast_BF <- ""
8 Data_Base <- ""
9 Query_fasta_file <- ""
10 Output_Path_ <- ""
11 #BLAST PARAMETRES
12 Output_file_name <- ""
13 word_size <- ""
14 Percentage_identity <- ""
15 number_of_threads <- ""
16 outputFormat <- "6"
17 Percentage_overlap <- ""
18 bitscore <- ""
19 InputFile_with_unique_ID <- ""
20
21 < -----
22
23 (Top Level) R Script
  
```

Figure 25 Example of window showing BLASTnFilter.R script.

- Fill the script and add the path to the Python executable, path to the BLAST+ bin directory, write TRUE if you are running a Discontinuous mega BLAST or FALSE if you are not, path to the BLAST+ database, path to the fasta file query, output path, output name, word size, percentage identity, number of threads, output format (only tabular format 6 is accepted), percentage overlap, bitscore, and path to file with Unique ID (Created in step 1).

The screenshot shows the RStudio interface with the BLASTnFilter.R script open. The code defines variables for Python executable, NCBI directory, Megablast binary, data base, query fasta file, output path, and various BLAST parameters like word size, percentage identity, number of threads, output format, percentage overlap, and bitscore. The 'filter.R' tab is also visible in the background.

```

1 #This script BLAST nucleotides sequences and filters
2 #
3 #PLEASE FILL THE FOLLOWING
4
5 python_exe <- "C:\\Users\\s433088\\AppData\\Local\\Programs\\Python\\Python38\\python.exe"
6 Path_to_NCBI_Directory <- "c:\\thingslap\\casual_work_2020\\blast-2.8.0+\\bin\\"
7 DC_MegaBlast_BF <- "FALSE"
8 Data_Base <- "C:\\thingslap\\Pipeline\\LINUX_Test\\E_faecium_genome\\E_faecium_genome.fasta"
9 Query_fasta_file <- "c:\\thingslap\\Pipeline\\LINUX_Test\\SampleData_Enterococcus_faecium_R.fasta"
10 Output_Path_ <- "c:\\thingslap\\Pipeline\\LINUX_Test\\"
11 #BLAST PARAMETERS
12 Output_file_name <- "Enterococcus_faecium_R_BLAST_"
13 word_size <- "11"
14 Percentage_identity <- "70"
15 number_of_threads <- "4"
16 OutputFormat <- "6"
17 Percentage_overlap <- "0.8"
18 bitscore <- "50"
19 InputFile_with_unique_ID <- "c:\\thingslap\\Pipeline\\LINUX_Test\\sampleData_Enterococcus_faecium_Inpu
20
21 ...
22 ...
23 ...
24 ...
25 ...
26 ...
27 ...
28 ...
29 ...
30 ...
31 ...
32 ...
33 ...
34 ...
35 ...
36 ...
37 ...
38 ...
39 ...
40 ...
41 ...
42 ...
43 ...
44 ...
45 ...
46 ...
47 ...
48 ...
49 ...
50 ...
51 ...
52 ...
53 ...
54 ...
55 ...
56 ...
57 ...
58 ...
59 ...
60 ...
61 ...
62 ...
63 ...
64 ...
65 ...

```

Figure 26 Example of script BLASTnFilter.R after filling parameters.

- Save the script, select all the code with CRTL+A, and run the code.

The screenshot shows the RStudio interface with the BLASTnFilter.R script running. The 'Console' tab displays the output of the BLAST search, showing hits for NZ_CP039729.1 against the Enterococcus faecium strain ZY2 genome. The 'Environment' and 'Files' panes are also visible.

```

#This script BLAST nucleotides sequences and filters
#
#PLEASE FILL THE FOLLOWING
#
python_exe <- "C:\\Users\\s433088\\AppData\\Local\\Programs\\Python\\Python38\\python.exe"
Path_to_NCBI_Directory <- "c:\\thingslap\\casual_work_2020\\blast-2.8.0+\\bin\\"
DC_MegaBlast_BF <- "FALSE"
Data_Base <- "C:\\thingslap\\Pipeline\\LINUX_Test\\E_faecium_genome\\E_faecium_genome.fasta"
Query_fasta_file <- "c:\\thingslap\\Pipeline\\LINUX_Test\\SampleData_Enterococcus_faecium_R.fasta"
Output_Path_ <- "c:\\thingslap\\Pipeline\\LINUX_Test\\"

```

```

994    NZ_CP039729.1  NZ_CP039729.1 Enterococcus faecium strain ZY2 chromosome, complete genome  TGCAGTTGAGGAAGAGAAAAAGAAGGTTTTGGTCACAAAAATCACCG  TGCAGTTGAGGAAAG
AAAAAGAAGGGTTTTGGTCACAAAATTACCG  46      4      92.000  50      3.14e-14
71.3   1      50      1986419  1986468  0      0      50      2736723  1.0
[1] 0
> |

```

Figure 27 Example of script BLASTnFilter.R after running the code.

This script creates 5 files.

| Name | Date modified | Type | Size |
|--|---------------------|---------------|----------|
| Enterococcus_faecium_R_BLAST_all_sequences_with_and_without_hits.txt | 25/08/2020 12:47 PM | Text Document | 425 KB |
| Enterococcus_faecium_R_BLAST_filtered.txt | 25/08/2020 12:47 PM | Text Document | 257 KB |
| Enterococcus_faecium_R_BLAST_only_sequences_with_hits.txt | 25/08/2020 12:47 PM | Text Document | 377 KB |
| Enterococcus_faecium_R_BLAST_sorted.txt | 25/08/2020 12:47 PM | Text Document | 215 KB |
| Enterococcus_faecium_R_BLAST_BLAST.txt | 25/08/2020 12:47 PM | Text Document | 3,057 KB |

Figure 28 Example of files created by BLASTnFilter.R script.

This step takes less than 5 minutes. The script produces 5 files if you included all inputs, or 3 files if you did not provide the File with UniqueID.

Description of files produced.

- **File 1. Enterococcus_faecium_R_BLAST_BLAST.txt**

This is the raw BLAST output. This file does not contain any headers and it is not filtered.

- **File 2. Enterococcus_faecium_R_BLAST_filtered.txt**

This file has headers and an extra column with the percentage overlap. Default filtering parameters for this tutorial are: Percentage Overlap >80%, bitscore >50, Percentage Identity>70. The percentage overlap can be modified according to the BLAST results expected. This file may contain multiple hits per sequence.

The BLAST has a table format 6 in the following order: "qseqid sacc stitle qseq sseq nident mismatch pident length evalue bitscore qstart qend sstart send gapopen gaps qlen slen". These are:

- qseqid: query (e.g., unknown gene) sequence id
- sacc: Subject accession
- stitle: Subject Title
- qseq: Aligned part of query sequence
- sseq: Aligned part of subject sequence
- nident: Number of identical matches
- mismatch: number of mismatches
- pident: percentage of identical matches
- length: alignment length (sequence overlap)
- evalue: expect value
- bitscore: bit score
- qstart: start of alignment in query
- qend: end of alignment in query
- sstart: start of alignment in subject
- send: end of alignment in subject
- gapopen: number of gap openings
- gaps: Total number of gaps
- qlen: Query sequence length
- slen: Subject sequence length

- **File 3. Enterococcus_faecium_R_BLAST_sorted.txt**

This file contains only one hit per sequence. The best match will be selected considering levels of sorting. First considering the highest percentage identity, then highest Percentage overlap, then highest bitscore and then taking only one Query per sequence.

If you didn't provide the file with UniqueID, this will be your final result.

- **File 4. Enterococcus_faecium_R_BLAST_only_sequences_with_hits.txt**
This file uses the UniqueID assigned to each sequence and writes the BLAST results back into the original file. This file only contains sequences that had a hit to something.
- **File 5. Enterococcus_faecium_R_BLAST_all_sequences_with_and_without_hits.txt**
This file uses the UniqueID assigned to each sequence and writes the BLAST results back into the original file. This file only contains all sequences with and without hits, written back into the original file.

5. Running BLAST and filtering script without the file with Unique ID.

If you do not have the file with Unique ID created with this pipeline, you can still run the BLAST and filtering. The steps are the same for running a normal BLAST but leave empty the string “InputFile_with_unique_ID” as shown in the picture.

```

#BLAST PARAMETERS
Output_file_name <- "Enterococcus_faecium_R_BLAST_Without"
word_size <- "11"
Percentage_identity <- "70"
number_of_threads <- "4"
OutputFormat <- "6"
Percentage_overlap <- "0.8"
bitscore <- "50"
InputFile_with_unique_ID <- "" # If you do not have this file, leave the string empty.

```

Running the code without the file will cause this error. It is normal.

```

Traceback (most recent call last):
  File "TestingPyCharm_BLAST_filtering_and_all.py", line 493, in <module>
    main(sys.argv[1:])
  File "TestingPyCharm_BLAST_filtering_and_all.py", line 442, in main
    Original_Modified_file_BF = open(Output_extra_file_BF, 'r')
FileNotFoundError: [Errno 2] No such file or directory: ''
[1] 1

```

Figure 29 Example of script BLASTnFilter.R after running the code without the file with UniqueID.

Running the code without this file will produce 4 text documents. Three of them with data and the final one empty. Your final file with the sequences of interest will have the termination “_sorted.txt”.

| Name | Date modified | Type | Size |
|--|--------------------|---------------|----------|
| Enterococcus_faecium_R_BLAST_Without_all_sequences_with_and_without_hits.txt | 25/08/2020 2:16 PM | Text Document | 0 KB |
| Enterococcus_faecium_R_BLAST_Without_sorted.txt | 25/08/2020 2:16 PM | Text Document | 215 KB |
| Enterococcus_faecium_R_BLAST_Without_filtered.txt | 25/08/2020 2:16 PM | Text Document | 257 KB |
| Enterococcus_faecium_R_BLAST_WithoutBLAST.txt | 25/08/2020 2:16 PM | Text Document | 3,057 KB |

Figure 30 Example output files generated after running the script BLASTnFilter.R without the file with UniqueID.

6. Additional filtering.

- For further filtering with different parameters without running BLAST+ again, you can use the script “filter.R”.
 - Clean the global environment before you run this script.

```
1 #This script filters a BLAST file
2 #
3 #PLEASE FILL THE FOLLOWING
4
5 python_exe <- "T"
6 a_BLAST_input_path_and_file_ <- ""
7 b_Output_Path_ <- ""
8 c_Output_file_name <- ""
9 d_Percentage_overlap <- ""
10 e_bitscore <- ""
11
12 ##### DO NOT MODIFY ANYTHING BELOW HERE#####
13 #####
14 #####
15 #####
16 #####
17 #
```

Figure 31 Example of filter.R script.

- Fill the script and add the path to the Python executable, path to the BLAST output file produced in the previous step, output path, output name, percentage overlap and bitscore.
 - This script can work on any BLAST output file which has exactly the same format as is used here. The format used here is BLAST output format 6, with the following columns: qseqid sacc stitle qseq sseq nident mismatch pident length evalue bitscore qstart qend sstart send gapopen gaps qlen slen.

Figure 32 Example of "filter.R script" after filling parameters.

- Save the script, select all the code with CRTL+A, and run the code.

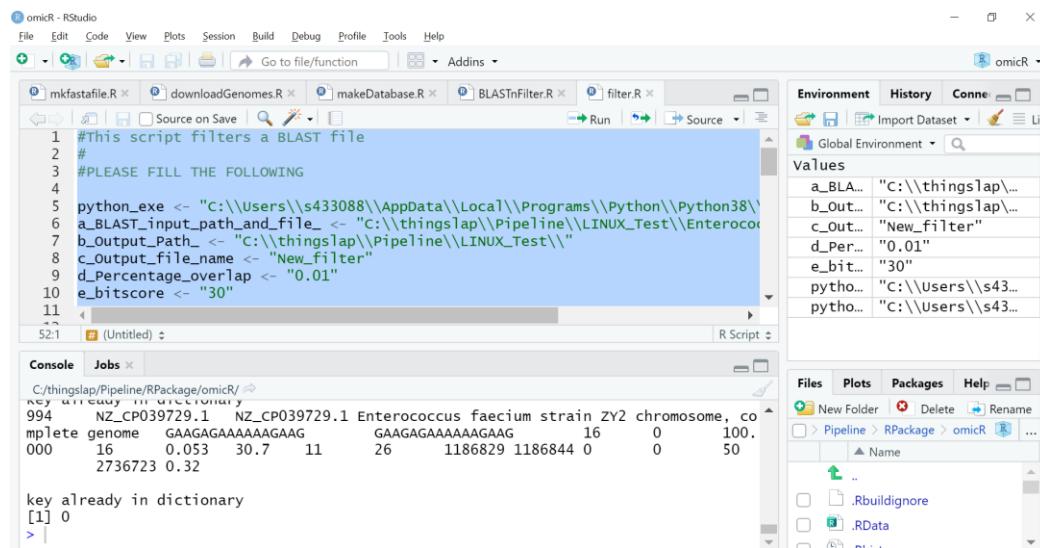


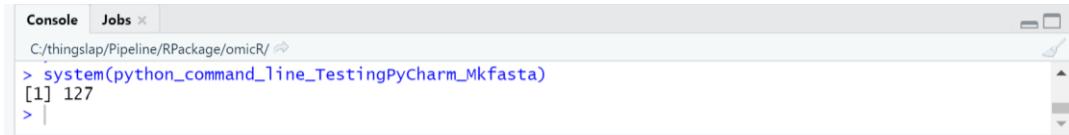
Figure 33 Example of "filter.R script" after running the code.

This will produce 2 files, with the new filtering parameters selected. The file with the name you provided and the ending “_sorted.txt” document is your final file.

| | | | |
|-------------------------|--------------------|---------------|--------|
| New_filter_filtered.txt | 25/08/2020 2:42 PM | Text Document | 424 KB |
| New_filter_sorted.txt | 25/08/2020 2:42 PM | Text Document | 240 KB |

7. Common errors

- If R is running, not producing any script and giving this error [1] 127, it means that the compiler is not found. Check that you are selecting the correct path to the Python executable and writing \\ in the path.



- 💣 Writing paths with a blank space between words.
 - 💣 Selecting the wrong file or wrong path.
 - 💣 Selecting “excel files” instead of “csv” files as input for initial step.
 - 💣 Not adding “\\” at the end of each path to a directory.

omicR for Linux

Download

https://github.com/BTalamantesBecerra/omicR_for_linux

Requirements

- NCBI BLAST+ V4 or latest. (<https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/>)
- Python V3 or latest (<https://www.python.org/downloads/>)
 - Package “tkinter”
- Biopython (<https://biopython.org/>)
 - Package “entrez”

Note: The files “omicR.py” and “Currito.xbm” need to be in the same directory.

Running instructions

Alternative 1:

- Open the script “omicR.py” using the IDLE or PyCharm and run it.
- This will open a window showing the Graphical User Interface where you can run the scripts.

Alternative 2:

- Open the script from the command line. Move directories to the location where you downloaded the script and run:

```
python3 omicR.py
```

- This will open a window showing the Graphical User Interface where you can run the scripts.

For usage, please refer to the section “Running “omicR” with graphical user interface” in this manual.

omicR for HPC computers

Download

https://github.com/BTalamantesBecerra/omicR_linux_commandline

Requirements

- NCBI BLAST+ V4 or latest. (<https://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/>)
- Python V3 or latest (<https://www.python.org/downloads/>)
- Biopython (<https://biopython.org/>)

If you are using Windows, I recommend downloading and installing Putty and WinSCP.

Add these programs to your environment path variables.

Introduction

If you are running omicR with an HPC computer, it likely that you know how to use a command line. For this purpose, I suggest that you only use 2 scripts to “create fasta files” and “filter”. As the steps of downloading, creating a database and running BLAST can take longer than running BLAST+ directly.

The required input BLAST command line to run this filtering script is:

```
blastn -db [ ] -query [ ] -out [ ] -word_size [ ] -perc_identity [ ] -num_threads [ ] -outfmt '6 qseqid sacc  
stitle qseq sseq nident mismatch pident length eval bitscore qstart qend sstart send gapopen gaps  
qlen slen'
```

In the following section, I will describe the steps for running all scripts.

STEPS

1. Create FASTA files and input files for BLAST / filtering.

Structure of the command line.

```
python3 (Call Python)  
TestingPyCharm_Mkfasta.py (Name of the Python script)  
-a (Path and name of Input CSV File)  
-b (Output path)  
-c (Name of output file)  
-d (Row with header- start counting from 1)  
-e (Row with data- start counting from 1)
```

-f (Column with sequence of interest - start counting from 0)

Sequence of interest: Column 4

| | A | B | C | D | E | F | G | H | I | J | K |
|----|----------|------------|-------------|---------|----------------|----------------|------------------|------------------|----------|----------|---|
| 1 | * | * | * | * | * | * | * | * | * | * | * |
| 2 | * | * | * | * | * | * | * | * | * | * | * |
| 3 | * | * | * | * | * | * | * | * | * | * | * |
| 4 | * | * | * | * | * | * | * | * | * | * | * |
| 5 | * | * | * | * | * | * | * | * | * | * | * |
| 6 | SeqIndex | ClusterIdx | ClusterSize | Tag | TrimmedSLength | LowComlChrom_d | ChromPosAInCnt_d | ChromPosAInValue | AlnCnt_d | AlnValue | |
| 7 | 89184 | 20963 | 408 | TGCAGAA | TGCAGAA | 49 | 0 | 0 | 0 | 999 | |
| 8 | 75556 | 20878 | 111 | TGCAGCG | TGCAGCG | 46 | 0 | 0 | 0 | 999 | |
| 9 | 73901 | 20862 | 96 | TGCAGTA | TGCAGTA | 41 | 0 | 0 | 0 | 999 | |
| 10 | 78622 | 20911 | 137 | TGCAGTT | TGCAGTT | 40 | 0 | 0 | 0 | 999 | |
| 11 | 78436 | 20902 | 127 | TGCAGCT | TGCAGCT | 43 | 0 | 0 | 0 | 999 | |
| 12 | 78175 | 20900 | 126 | TGCAGCA | TGCAGCA | 46 | 0 | 0 | 0 | 999 | |
| 13 | 79343 | 20909 | 134 | TGCAGAT | TGCAGAT | 52 | 0 | 0 | 0 | 999 | |

Figure 34 Display of how rows and headers should be selected.

An example of how the command line should look is below:

```
Python3      TestingPyCharm_Mkfasta.py      -a      ~Path\YourFile.csv      -b
~Path\Directory\ -c YourFileName -d 7 -e 8 -f 2
```

Note: Before running this command line, you must be in the location where the script is saved to be able to run it.

Move directories until you get to the location where the Python script was saved.

This is an example of how it should look

```
[talamantes@dungog data/scratch/test_talamantes/Python_tutorial]$
[talamantes@dungog Python_tutorial]$ python3 TestingPyCharm_Mkfasta.py -a /data/scratch/test_talamantes/Python_tutorial/SampleData_Enterococcus_faecium.csv -b /data/scratch/test_talamantes/Python_tutorial/Outputs/ -c Enterococcus_faecium -d 6 -e 7 -f 4
[talamantes@dungog Python_tutorial]$
```

The two output files should be in the Outputs directory:

```
[talamantes@dungog Outputs]$ ls
Enterococcus_faecium.fasta  Enterococcus_faecium_InputFile_with_unique_ID.txt
[talamantes@dungog Outputs]$
```

2. Download genomes from the NCBI.

Structure of the command line.

python3 (Call Python)

TestingPyCharm_Downloading_genomes.py (Name of the Python script)

-a (your email)

-b (RefSeq number)

-c (Path for outputs)

-d (Name of output file)

An example of how the command line should look is given below:

```
Python3 TestingPyCharm_Downloading_genomes.py -a  
youremail@hotmail.com -b GCF_900067755.1 -c ~Path\Directory\ -d  
YourGenomeName
```

This script is recommended for small genomes or chromosomes. If you are planning to download large genomes from NCBI, it is recommended to use the browser to download from the NCBI website. You cannot modify the speed of fetching in this step. For example, a bacterial genome such as NZ_CP039730.1 takes a few seconds, but the entire chicken genome can take approximately 5 hours.

You need to provide your email as NCBI needs to identify you for granting you access, otherwise your access can be denied.

Note: Before running this command line, you must be in the location where the script is saved to be able to run it.

Move directories until you get to the location where the Python script was saved.

3. Create NCBI database for BLAST+

Structure of the command line.

python3 (Call Python)

TestingPyCharm_MakeDataBase.py (Name of the Python script)

-a Path to your BLAST+ bin directory

-b (Path to your genome file as fasta or fna file)

-c (Type of database, use nucl for nucleotides)

An example of how the command line should look is given below:

```
python3 TestingPyCharm_MakeDataBase.py -a ~PATH\blast-2.8.0+\bin\ -b ~Path\MyGenome.fna -c  
nucl
```

Notes:

* Before running this command line, you must be in the location where the script is saved to be able to run it.

* Move directories until you get to the location where the Python script was saved.

* If you already have the mkblastdb in the global environment paths do not include the handle '-a'

If it works, it should look like this:

```
Building a new DB, current time: 08/25/2020 16:59:51
New DB name: /data/scratch/test_talamantes/Python_tutorial/Outputs/E_faecium_genome/E_faecium_genome.fasta
New DB title: /data/scratch/test_talamantes/Python_tutorial/Outputs/E_faecium_genome/E_faecium_genome.fasta
Sequence type: Nucleotide
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 2 sequences in 0.029314 seconds.
[talamantes@dungog Python_tutorial]$ dir
Outputs          TestingPyCharm_BLAST_filtering_and_all.py  TestingPyCharm_MakeDataBase.py  TestingPyCharm_NCBI_BLAST_filtering.py
SampleData_Enterococcus_faecium.csv  TestingPyCharm_Downloading_genomes.py  TestingPyCharm_MKfasta.py
[talamantes@dungog Python_tutorial]$ cd Outputs/
[talamantes@dungog Outputs]$ ls
E_faecium_genome  Enterococcus_faecium.fasta  Enterococcus_faecium_InputFile_with_unique_ID.txt
[talamantes@dungog Outputs]$ cd E_faecium_genome/
[talamantes@dungog E_faecium_genome]$ ls
E_faecium_genome.fasta  E_faecium_genome.fasta.nhr  E_faecium_genome.fasta.nin  E_faecium_genome.fasta.nsq
[talamantes@dungog E_faecium_genome]$ █
```

4. BLAST and filtering.

Structure of the command line:

python3 (Call Python)

TestingPyCharm_BLAST_filtering_and_all.py (Name of the Python script)

-y (**TRUE** if you are running a Dissimilar discontinuous MegaBLAST or **FALSE** if you are not)

-a (Path to the reference genome)

-b (Path to the query sequences)

-c (Path to the output directory)

-d (Name of output directory)

-e (Word Size: recommended value 11)

-f (Percentage identity: recommended value 70)

-g (Threads)

-i (Output format, only tabular format 6 is accepted for this script)

-j (Percentage overlap: recommended value 0.8)

-k (Bitscore: recommended value 50)

An example of how the command line should look like is below:

```
python3 TestingPyCharm_BLAST_filtering_and_all.py -y FALSE -a  
Path/MyGenome.fna -b Path/MySequences.fasta -c ~Path/directory/ -d  
MyBlastResults -e 11 -f 70 -g 35 -i 6 -j 0.8 -k 50 -l
```

Notes: Before running this command line, you must be in the location where the script is saved to be able to run it.

***Move directories until you get to the location where the Python script was saved.**

If you ran the command line without including the file with unique ID, one file without data will be generated “Enterococcus_faecium_Results_all_sequences_with_and_without_hits.txt”. This is normal. Your final file has the name “Enterococcus_faecium_Results_sorted.txt”

If you ran these two command lines, your outputs will be similar to this:

The outputs are here →

```

talamantes@clungog:/data/scratch/test_talamantes/Python_tutorial/Outputs
TATTACATGCAAAATGGAAAGCTGTCTTGGAAATCCG 45 1 97.826 46 4.56e-17 80.5 1 46 2415927 2415972 0 0 46 2
736723 1.0

987 NZ CP039729.1 NZ CP039729.1 Enterococcus faecium strain ZY2 chromosome, complete genome TGCAGTATTACATGCAAAATGGAAAGCGCTCTTTGGAAATCCG TGCAG
TAITTCAGATGCAAAATGGAAAGCTGTCTTGGAAATCCG 45 1 97.826 46 4.56e-17 80.5 1 46 2415927 2415972 0 0 46 2
736723 1.0

988 NZ CP039729.1 NZ CP039729.1 Enterococcus faecium strain ZY2 chromosome, complete genome TGCAGCATCGCTCTGAAGACTAGGGGTACGTATTAGAACCG TGCAG
CATCGCTTGAAGAACTAGGCCGTTACGTATTAGAACCG 45 1 97.826 46 4.56e-17 80.5 1 46 2354284 2354329 0 0 46 2
736723 1.0

989 NZ CP039729.1 NZ CP039729.1 Enterococcus faecium strain ZY2 chromosome, complete genome TGCAGCATCGCTCTGAAGACTAGGGGTACGTATTAGAACCG TGCAG
CATCGCTTGAAGAACTAGGCCGTTACGTATTAGAACCG 45 1 97.826 46 4.56e-17 80.5 1 46 2354284 2354329 0 0 46 2
736723 1.0

99 NZ CP039729.1 NZ CP039729.1 Enterococcus faecium strain ZY2 chromosome, complete genome TGCAGTTGAGGAAAAGAAAAAGAGGGTTTGGTCACAAAAATTACCG T
ACAGTTGAGGAAAAGAAAAAGAGGGTTTGGTCACAAAAATTACCG 50 0 100.000 50 6.69e-21 93.5 1 50 1986419 1986468 0 0 5
0 2736723 1.0

990 NZ CP039729.1 NZ CP039729.1 Enterococcus faecium strain ZY2 chromosome, complete genome TGCAGCATCGCTCTGAAGACTAGGGGTACGTACTAGAACCG TGCAG
CATCGCTTGAAGAACTAGGCCGTTACGTATTAGAACCG 45 1 97.826 46 4.56e-17 80.5 1 46 2354284 2354329 0 0 46 2
736723 1.0

991 NZ CP039729.1 NZ CP039729.1 Enterococcus faecium strain ZY2 chromosome, complete genome GCAGAACATAATGCTGGATTATTGTCATTATATGATGT GCAAGA
ACATAATGTTGGATTAACGTACATGATGT 38 3 92.683 41 6.15e-11 60.2 2 42 0 47 27367
23 0.8723404255319149 0 0 47 27367

993 NZ CP039729.1 NZ CP039729.1 Enterococcus faecium strain ZY2 chromosome, complete genome TGCAGTTGAGGAAAAGAGAAAAGAGGGTTTGGTCACAAAAATTACCG T
ACAGTTGAGGAAAAGAAAAAGAGGGTTTGGTCACAAAAATTACCG 46 4 92.000 50 3.14e-14 71.3 1 50 1986419 1986468 0 0 5
0 2736723 1.0

994 NZ CP039729.1 NZ CP039729.1 Enterococcus faecium strain ZY2 chromosome, complete genome TGCAGTTGAGGAAAAGAGAAAAGAGGGTTTGGTCACAAAAATTACCG T
ACAGTTGAGGAAAAGAAAAAGAGGGTTTGGTCACAAAAATTACCG 46 4 92.000 50 3.14e-14 71.3 1 50 1986419 1986468 0 0 5
0 2736723 1.0

[talamantes@dungog Python_tutorial]$ cd Outputs/
[talamantes@dungog Outputs]$ ls
E.faecium_genome Enterococcus_faecium_only_sequences_with_hits.txt
Enterococcus_faecium_all_sequences_with_and_without_hits.txt Enterococcus_faecium_Results_all_sequences_with_and_without_hits.txt
Enterococcus_faecium_Blast.txt Enterococcus_faecium_ResultsBlast.txt
Enterococcus_faecium.fasta Enterococcus_faecium_Results_filtered.txt
Enterococcus_faecium_filtered.txt Enterococcus_faecium_Results_sorted.txt
Enterococcus_faecium_Inputfile_with_unique_ID.txt Enterococcus_faecium_sorted.txt
[talamantes@dungog Outputs]$
```

It will produce the 5 files, or the 3 files if you did not provide the File with UniqueID.

Description of files produced.

- File 1. Enterococcus_faecium_BLAST.txt**

This is the raw BLAST output. This file does not contain any headers and it is not filtered.

- File 2. Enterococcus_faecium_filtered.txt**

This file has headers and an extra column with the percentage overlap. Default filtering parameters for this tutorial are: Percentage Overlap >80%, bitscore >50, Percentage Identity>70. The percentage overlap can be modified according to the BLAST results expected. This file may contain multiple hits per sequence.

The BLAST has a table format 6 in the following order: "qseqid sacc stitle qseq sseq nident mismatch pident length evalue bitscore qstart qend sstart send gapopen gaps qlen slen". These are:

- qseqid: query (e.g., unknown gene) sequence id
- sacc: Subject accession
- stitle: Subject Title
- qseq: Aligned part of query sequence
- sseq: Aligned part of subject sequence
- nident: Number of identical matches
- mismatch: number of mismatches
- pident: percentage of identical matches
- length: alignment length (sequence overlap)
- evalue: expect value
- bitscore: bit score
- qstart: start of alignment in query

- qend: end of alignment in query
 - sstart: start of alignment in subject
 - send: end of alignment in subject
 - gapopen: number of gap openings
 - gaps: Total number of gaps
 - qlen: Query sequence length
 - slen: Subject sequence length
- **File 3. Enterococcus_faecium_sorted.txt**
 This file contains only one hit per sequence. The best match will be selected considering levels of sorting. First considering the highest percentage identity, then highest Percentage overlap, then highest bitscore and then taking only one Query per sequence.
 If you didn't provide the file with UniqueID, this will be your final result.
- **File 4. Enterococcus_faecium_only_sequences_with_hits.txt**
 This file uses the UniqueID assigned to each sequence and writes the BLAST results back into the original file. This file only contains sequences that had a hit to something.
- **File 5. Enterococcus_faecium_all_sequences_with_and_without_hits.txt**
 This file uses the UniqueID assigned to each sequence and writes the BLAST results back into the original file. This file only contains all sequences with and without hits, written back into the original file.

5. Additional filtering.

Structure of the command line:

```
python3 (Call Python)
TestingPyCharm_NCBI_BLAST_filtering.py (Name of the Python script)
-a (Path and name of BLAST output file)
-b (Path to the output directory)
-c (Name of output directory)
-d (Percentage overlap: recommended value 0.8)
-e (Bitscore: recommended value 50)
```

An example of how the command line should look like is below:

```
python3 TestingPyCharm_NCBI_BLAST_filtering.py -a Path/MyBLAST_Results.txt
-b ~Path/directory/ -c My_Filtered_Blast_Results -d 0.8 -e 50
```

Note: Before running this command line, you must be in the location where the script is saved to be able to run it.

Move directories until you get to the location where the Python script was saved.

***You have to run the previous step (or command lines) to be able to run this command line.**

If you run it, the screen will look like this.

| talamanter@dungooy:/data/scratch/test_talamanter/Python.tutorial | | | | | | | | | | | |
|--|--|---------------|---------------------------------|-----------------------------|--|--|---|----|---------|---------|--------|
| 984 | NZ_CP039730.1 | NZ_CP039730.1 | Enterococcus faecium strain ZY2 | plasmid pZY2 | CCATAATACGGGGATAACGACTGTATGACGTGAAACC | CCATCATACGTGGATAACGGCTATGA | | | | | |
| | GTTGAAACC | 34 | 3 | 91.892 37 | 9.94e-09 | 52.8 | 9 | 45 | 44268 | 44304 | 0 0 46 |
| 985 | NZ_CP039730.1 | NZ_CP039730.1 | Enterococcus faecium strain ZY2 | plasmid pZY2 | CCATCATACGGGGATAACAACTGTATGACGTGAAACC | CCATCATACGTGGATAACGGCTATGA | | | | | |
| | GTTGAAACC | 34 | 3 | 91.892 37 | 9.94e-09 | 52.8 | 9 | 45 | 44268 | 44304 | 0 0 46 |
| 986 | NZ_CP039729.1 | NZ_CP039729.1 | Enterococcus faecium strain ZY2 | chromosome, complete genome | TGCAGTATTACAACGACAATGGAAAGCTGTCTTTGGGATCCG | TGCAGTATTACAACGACAATGGAAAGCTGTCTTTGGGATCCG | | | | | |
| | TATTTACAATGACAATGGAAAGCTGTCTTTGGGATCCG | 45 | 1 | 97.826 46 | 4.56e-17 | 80.5 | 1 | 46 | 2415927 | 2415972 | 0 0 46 |
| 736723 | 1.0 | | | | | | | | | | |
| 987 | NZ_CP039729.1 | NZ_CP039729.1 | Enterococcus faecium strain ZY2 | chromosome, complete genome | TGCAGTATTACAATGACAATGGAAAGCTGTCTTTGGGATCCG | TGCAGTATTACAATGACAATGGAAAGCTGTCTTTGGGATCCG | | | | | |
| | TATTTACAATGACAATGGAAAGCTGTCTTTGGGATCCG | 45 | 1 | 97.826 46 | 4.56e-17 | 80.5 | 1 | 46 | 2415927 | 2415972 | 0 0 46 |
| 736723 | 1.0 | | | | | | | | | | |
| 988 | NZ_CP039729.1 | NZ_CP039729.1 | Enterococcus faecium strain ZY2 | chromosome, complete genome | TGCAGCATTGAGAACTAGGCCGTTACGTATTAGAACCG | TGCAGCATTGAGAACTAGGCCGTTACGTATTAGAACCG | | | | | |
| | CATCGCTTGAAGAACTAGGCCGTTACGTATTAGAACCG | 45 | 1 | 97.826 46 | 4.56e-17 | 80.5 | 1 | 46 | 2354284 | 2354329 | 0 0 46 |
| 736723 | 1.0 | | | | | | | | | | |
| 989 | NZ_CP039729.1 | NZ_CP039729.1 | Enterococcus faecium strain ZY2 | chromosome, complete genome | TGCAGCATTGAGAACTAGGCCGTTACGTATTAGAACCG | TGCAGCATTGAGAACTAGGCCGTTACGTATTAGAACCG | | | | | |
| | CATCGCTTGAAGAACTAGGCCGTTACGTATTAGAACCG | 45 | 1 | 97.826 46 | 4.56e-17 | 80.5 | 1 | 46 | 2354284 | 2354329 | 0 0 46 |
| 736723 | 1.0 | | | | | | | | | | |
| 990 | NZ_CP039729.1 | NZ_CP039729.1 | Enterococcus faecium strain ZY2 | chromosome, complete genome | TGCAGTGGAAAAGAAAAAGAGGTTTGTCACAAAAATTACCG | TGCAGTGGAAAAGAAAAAGAGGTTTGTCACAAAAATTACCG | | | | | |
| | GCAGTGGAAAAGAAAAAGAGGTTTGTCACAAAAATTACCG | 50 | 0 | 100.000 50 | 6.69e-21 | 93.5 | 1 | 50 | 1986419 | 1986468 | 0 0 |
| 0 | 2736723 | 1.0 | | | | | | | | | |
| 991 | NZ_CP039729.1 | NZ_CP039729.1 | Enterococcus faecium strain ZY2 | chromosome, complete genome | GCAGTGGAAAAGAAAAAGAGGTTTGTCACAAAAATTACCG | GCAGTGGAAAAGAAAAAGAGGTTTGTCACAAAAATTACCG | | | | | |
| | ACAAATGTTGGATTACTCTTAAATAGATGT | 38 | 3 | 92.683 41 | 6.15e-11 | 60.2 | 2 | 42 | 855172 | 855212 | 0 0 47 |
| 23 | 0.8723404255319149 | | | | | | | | | | |
| 993 | NZ_CP039729.1 | NZ_CP039729.1 | Enterococcus faecium strain ZY2 | chromosome, complete genome | TGCAGTGGAAAAGAAAAAGAGGTTTGTCACAAAAATTACCG | TGCAGTGGAAAAGAAAAAGAGGTTTGTCACAAAAATTACCG | | | | | |
| | GCAGTGGAAAAGAAAAAGAGGTTTGTCACAAAAATTACCG | 46 | 4 | 92.000 50 | 3.14e-14 | 71.3 | 1 | 50 | 1986419 | 1986468 | 0 0 |
| 0 | 2736723 | 1.0 | | | | | | | | | |
| 994 | NZ_CP039729.1 | NZ_CP039729.1 | Enterococcus faecium strain ZY2 | chromosome, complete genome | TGCAGTGGAAAAGAAAAAGAGGTTTGTCACAAAAATTACCG | TGCAGTGGAAAAGAAAAAGAGGTTTGTCACAAAAATTACCG | | | | | |
| | GCAGTGGAAAAGAAAAAGAGGTTTGTCACAAAAATTACCG | 46 | 4 | 92.000 50 | 3.14e-14 | 71.3 | 1 | 50 | 1986419 | 1986468 | 0 0 |
| 0 | 2736723 | 1.0 | | | | | | | | | |

The two new files will be in the output directory with the other files.

The End